

# Accessing the mainframe as a “server”

# Accessing the Mainframe

## **Past:**

- Many applications are tied to z/OS (CICS, DB2)
- New developments made on other platforms

## **Now:**

- Integrate both on z/OS

# Using the MF as a server

**Two basic ways to use the mainframe as a server:**

- MQ - messaging via queues
- Web Services - HTTP, J2EE

# Using the MF as a server - MQ

## **Two basic ways to use the mainframe as a server:**

- MQ - messaging via queues
  - Messaged-queue based communication
  - Assured delivery: messages do not get lost and they arrive only once
  - No synchronous access needed
  - Message driven application
  - Quicker development due to shielding of the network
  - Available on multiple platforms
  - On z/OS it has interfacing to CICS, IMS, Batch or TSO

# Using the MF as a server - Web Services

**Two basic ways to use the mainframe as a server:**

- Web Services - HTTP, J2EE
  - EBCDIC/ASCII file access
  - System Management Facilities
  - Tracing & logging
  - Server Side Includes
  - Cookies support
  - Multi Format Processing
  - Persistent connections
  - Virtual hosts
  - Security

# Access using MQ

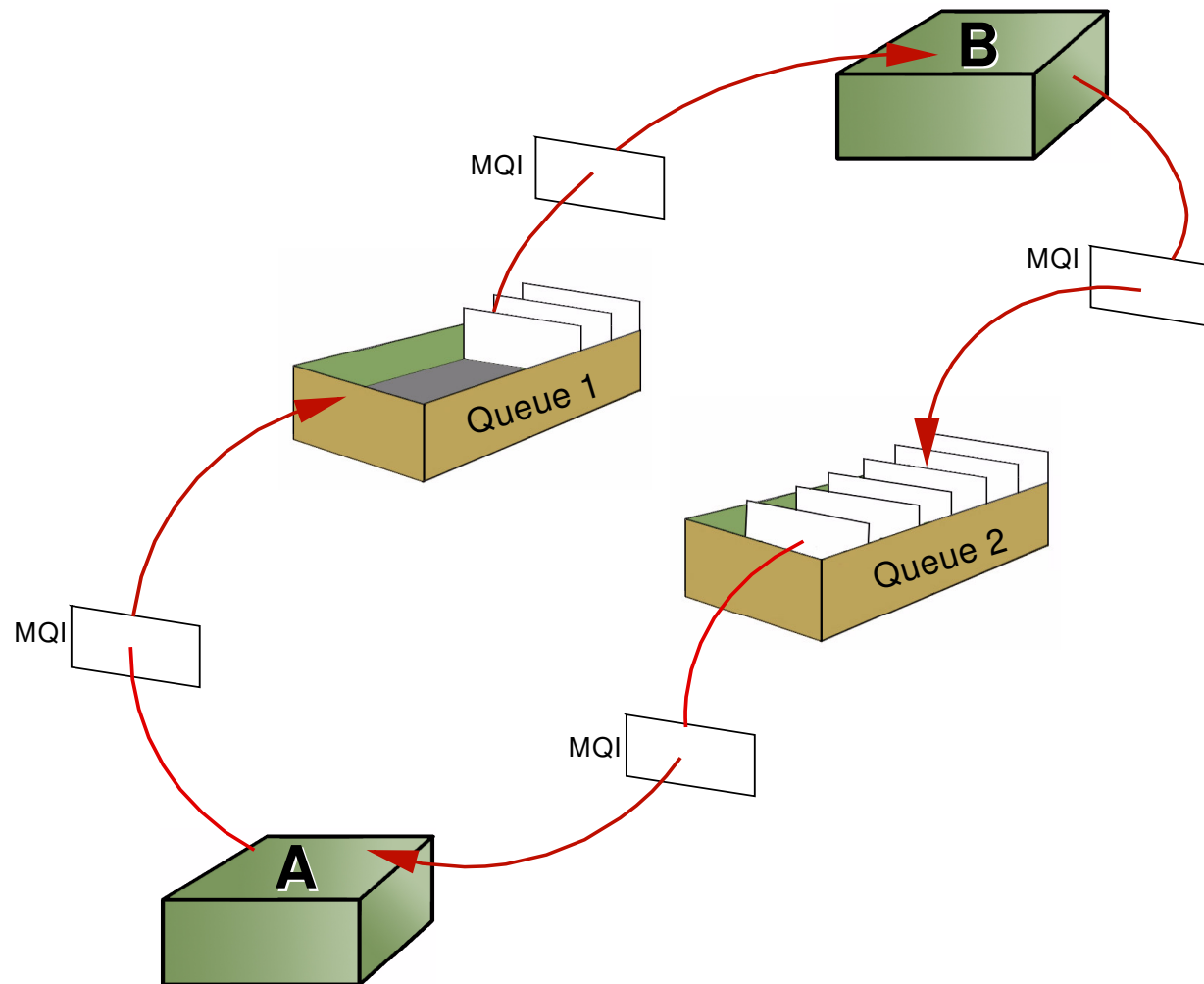
# Access using MQ

## **MQ is a Multi-platform messaging system**

- Available on all major platforms
- Takes care of data conversions & communications
- Enables a web server running on a platform to access CICS transactions running on another platform.

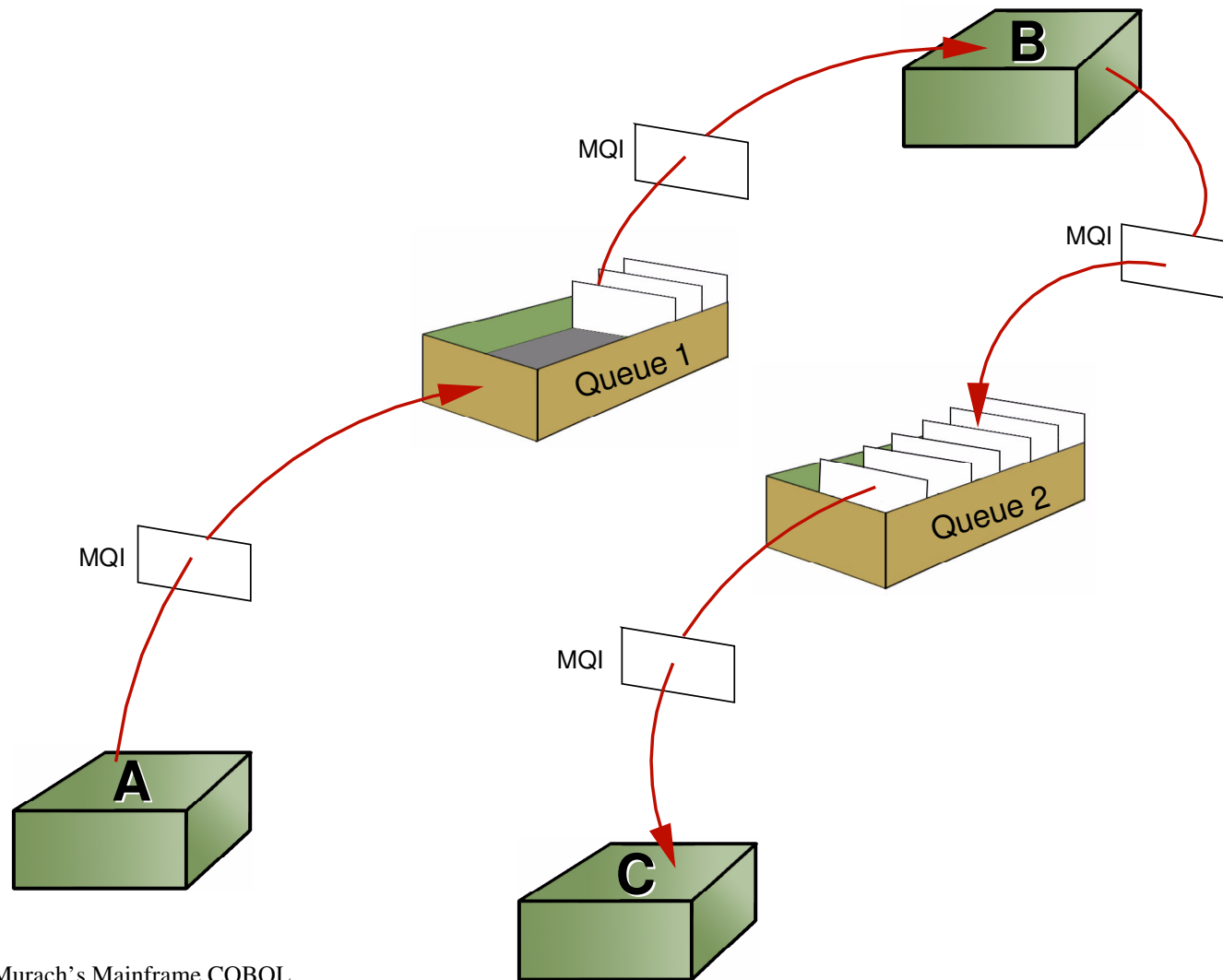
An “MQ CICS Bridge” enables one to access (via MQ) a 3270 transaction (and get the response back via MQ)

# MQ Synchronous communication model

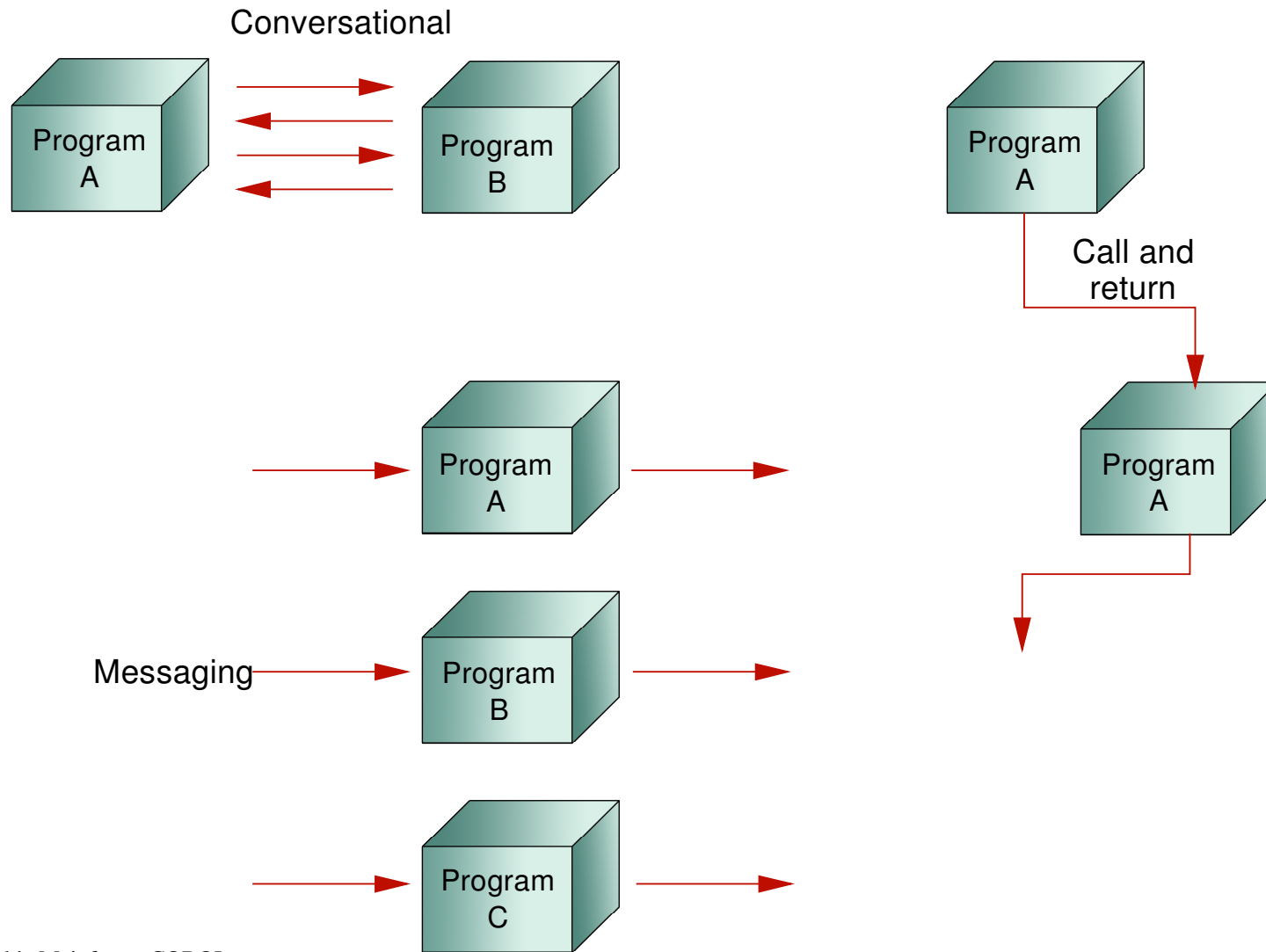




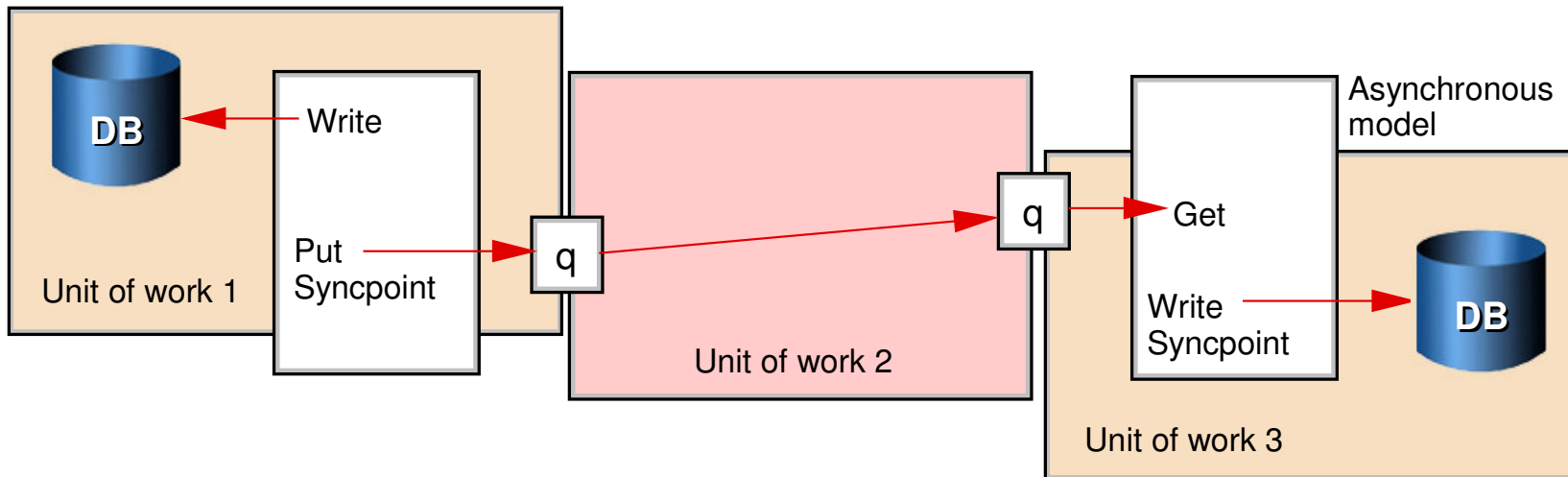
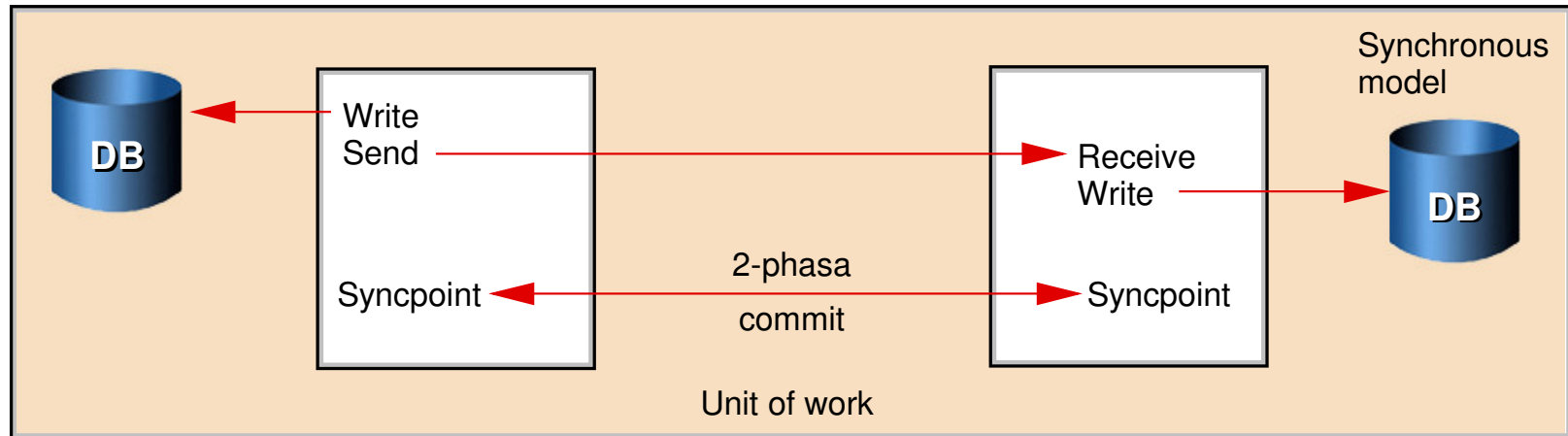
# MQ Asynchronous communication model



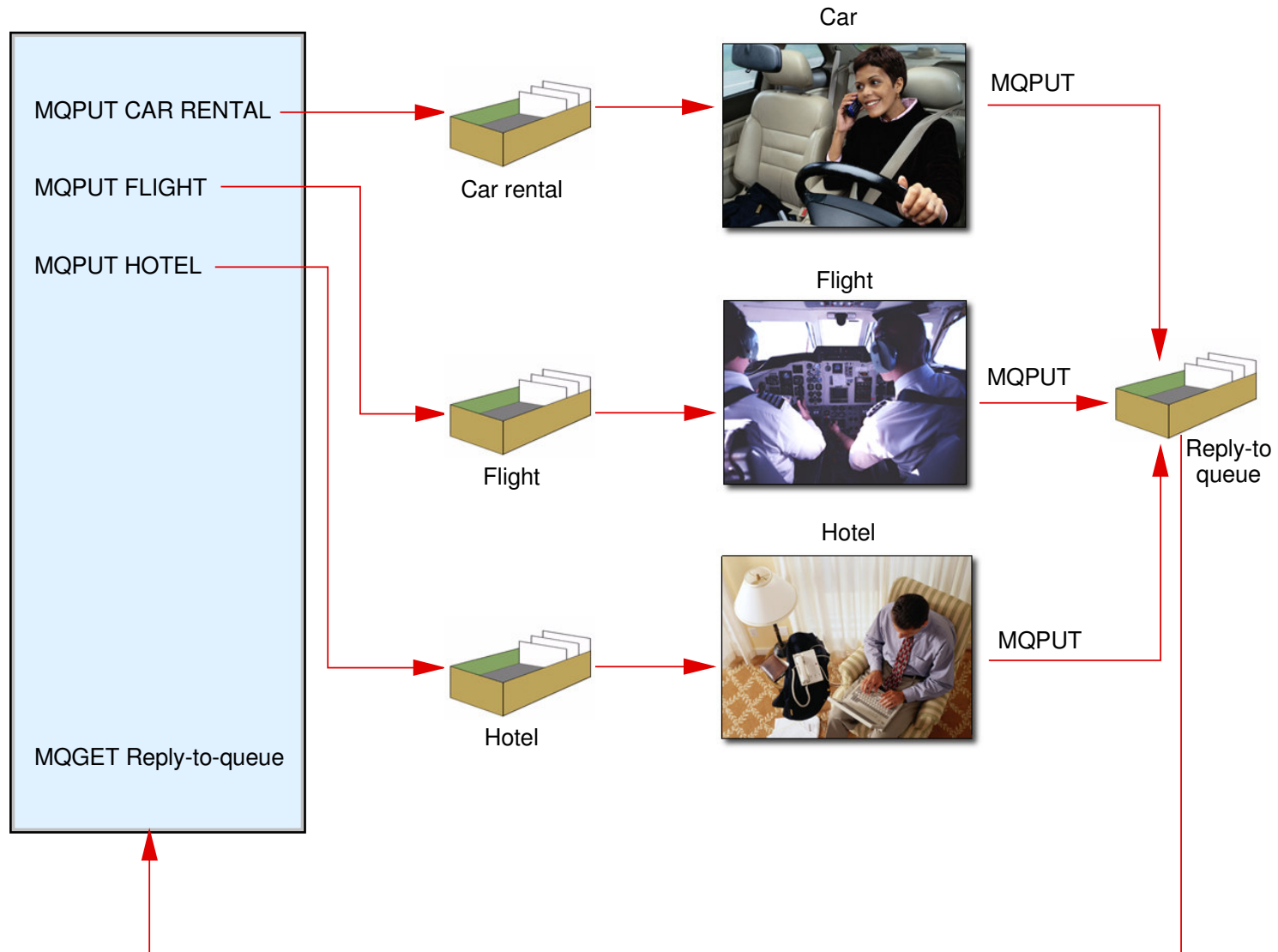
# MQ: Styles of Communication



# Data Integrity



# Travel agency example



# Access using Web Services

# HTTP server capabilities

## **Security functions:**

- Thread level security
- HTTPS/SSL support
- LDAP support
- Certificate authentication
- Proxy support

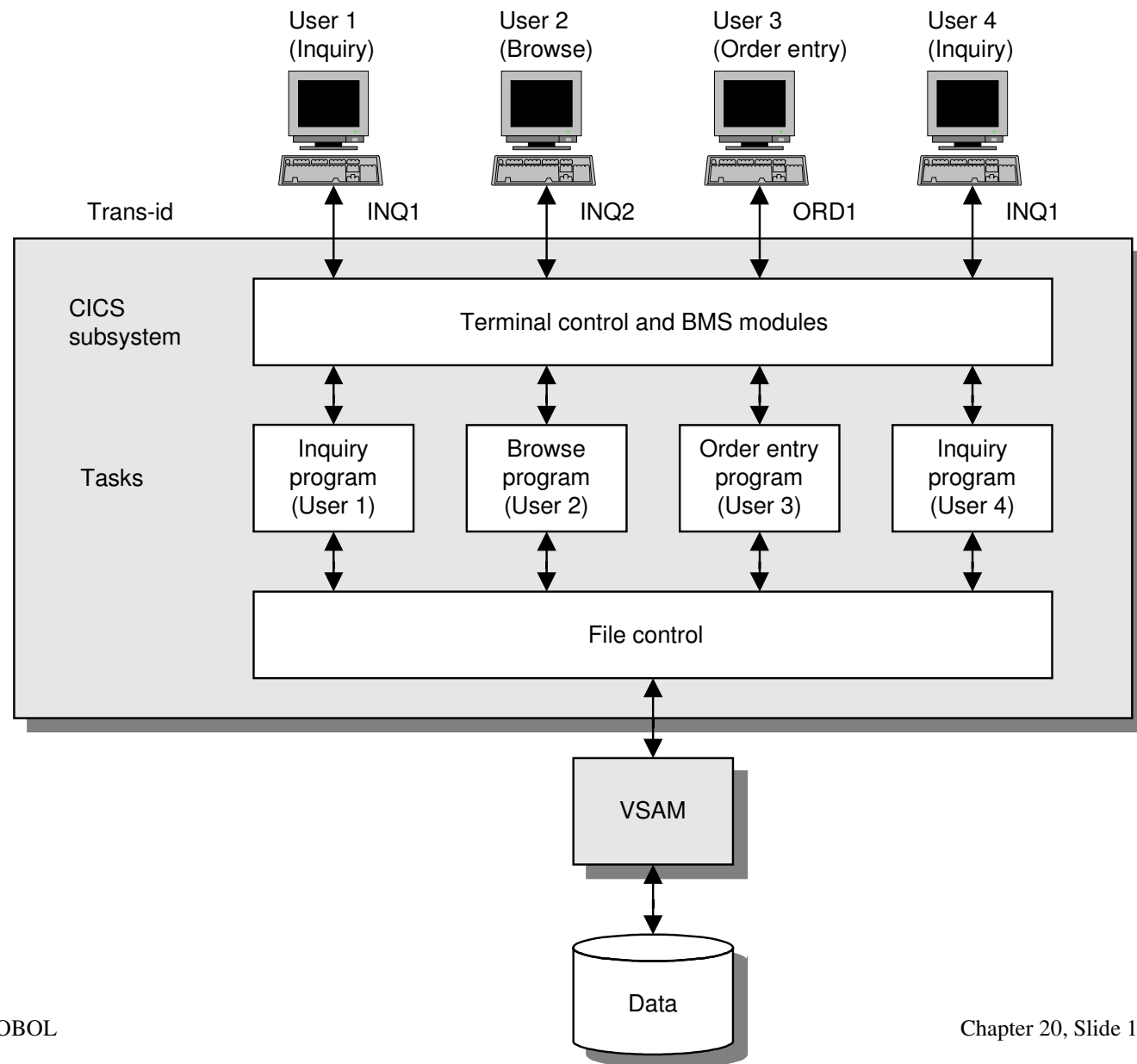
## **File caching:**

- HTTP server caching HFS files
- HTTP server caching z/OS data sets
- Unix System Service caching HFS files
- Fast Response Cache Accelerator (FRCA)

# Z/OS HTTP Server

- **Server modes:**
  - Stand-alone server
  - Scalable server
  - Multiple servers
- **Static Web pages**
- **Dynamic Web pages**

# Reviewing CICS services





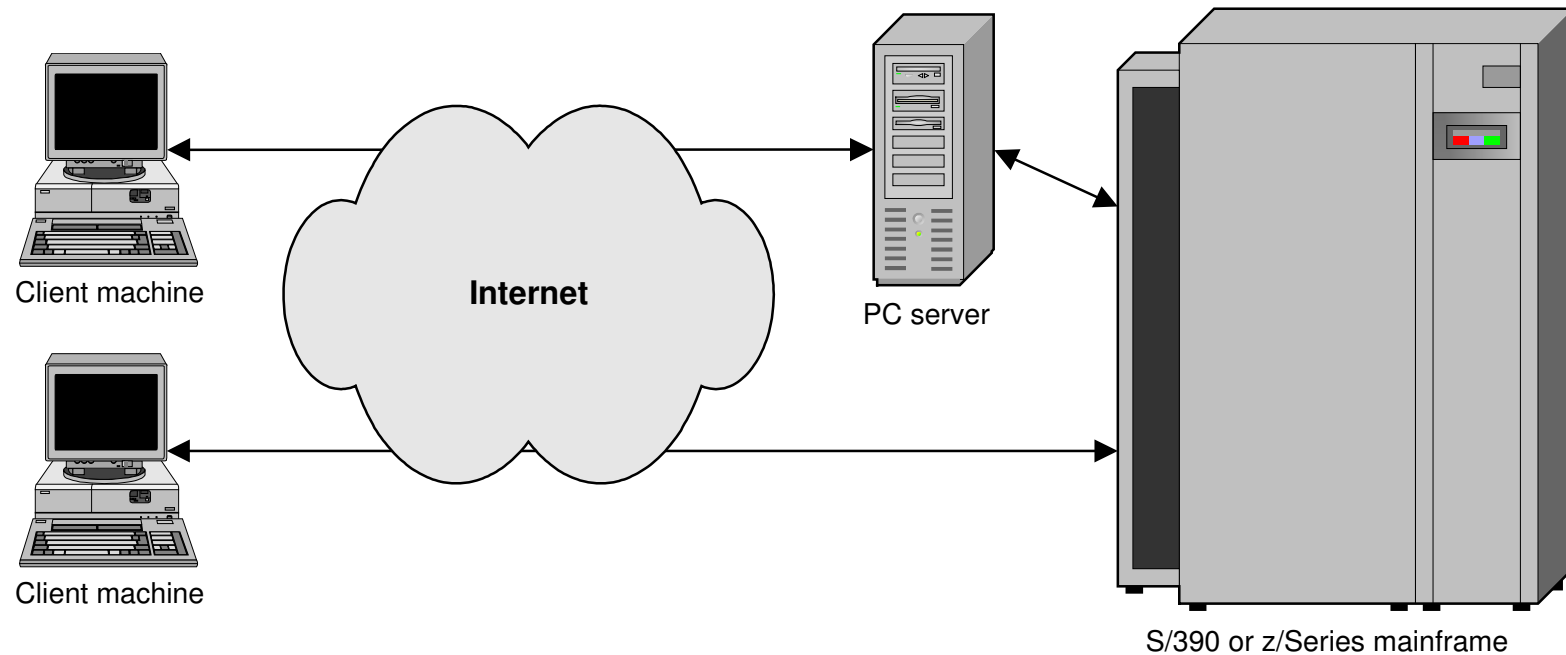
## Three benefits of using CICS with web applications

- CICS is especially designed for handling a large number of users and a large number of transactions while insuring the integrity of the data being processed.
- A large number of mission-critical business applications are already written in CICS.
- It's easier and less expensive to web-enable a COBOL/CICS application than it is to rewrite it in another language for another platform.

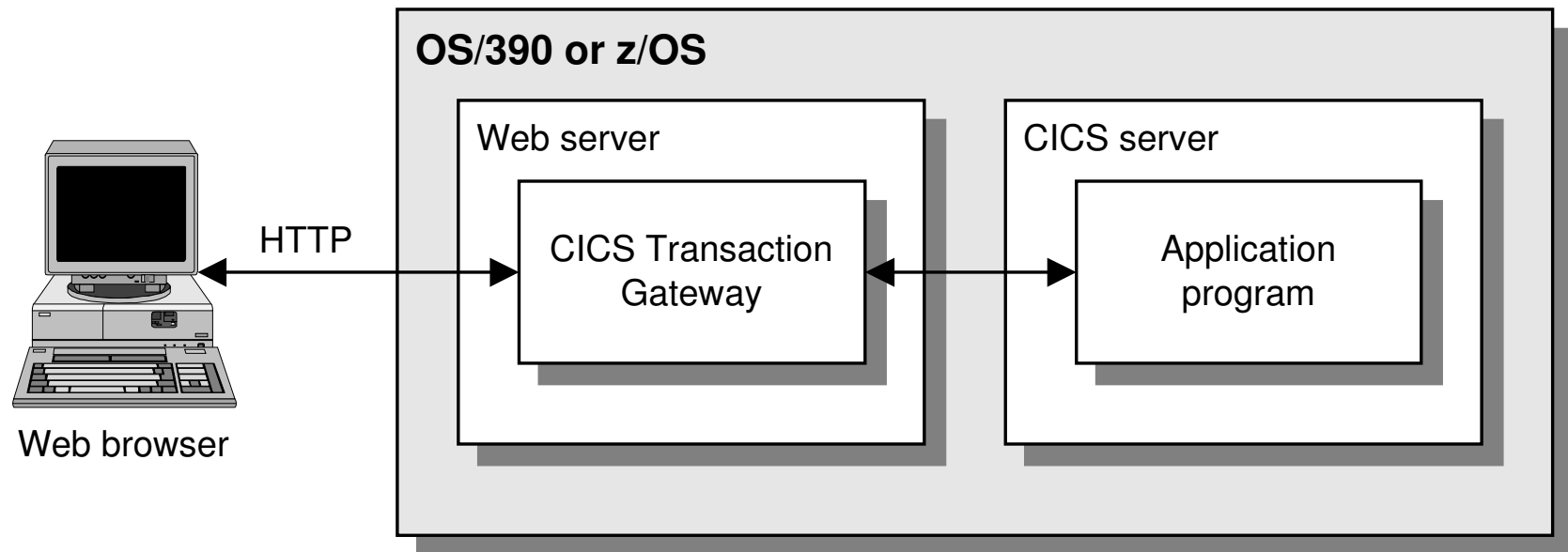
## CICS and the Internet

- An Internet user can access a CICS application running on a mainframe directly or through an intermediate server such as a PC server.
- To access the Internet, a *web browser* is run on the client machine.
- The main job of the browser is to process pages written in *HTML* (*Hypertext Markup Language*).
- The browser communicates with the mainframe or intermediate server using *HTTP* (*Hypertext Transfer Protocol*).
- The mainframe or intermediate server must provide web server software that interprets the HTTP requests sent from the user's web browser and translates them into a form that CICS recognizes.

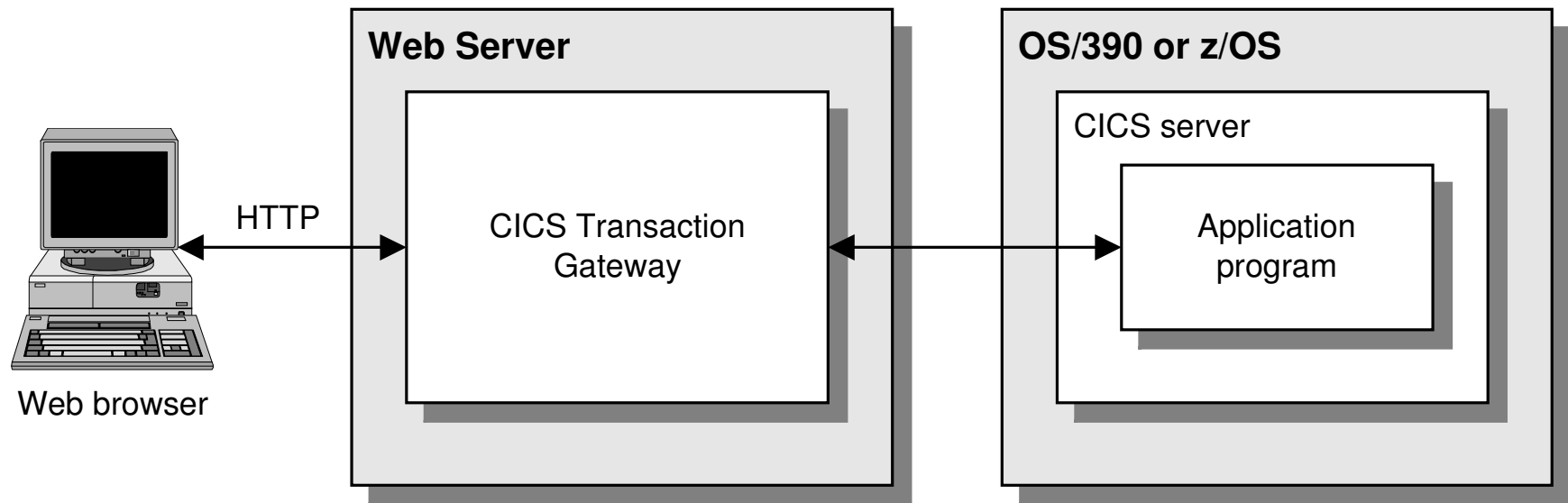
## Two ways to connect a client machine to a mainframe through the Internet



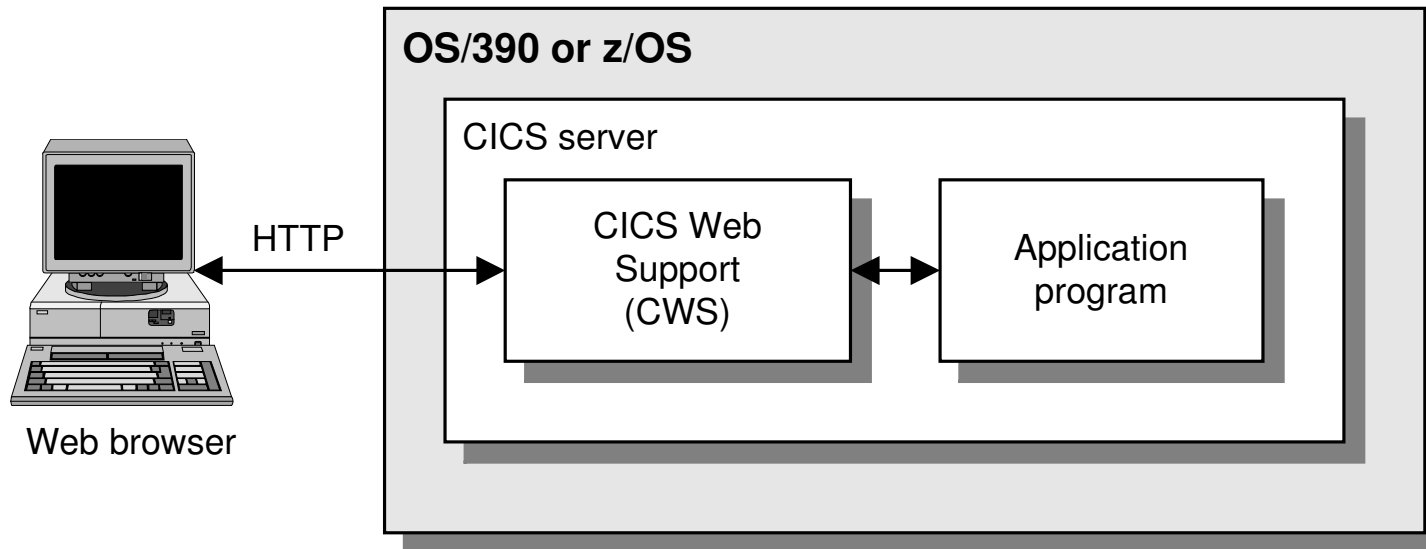
# How a web server running under OS/390 or z/OS accesses CICS programs



# How a web server running on a separate system accesses CICS programs



## How CICS provides for web support on its own



# Access using Web Services

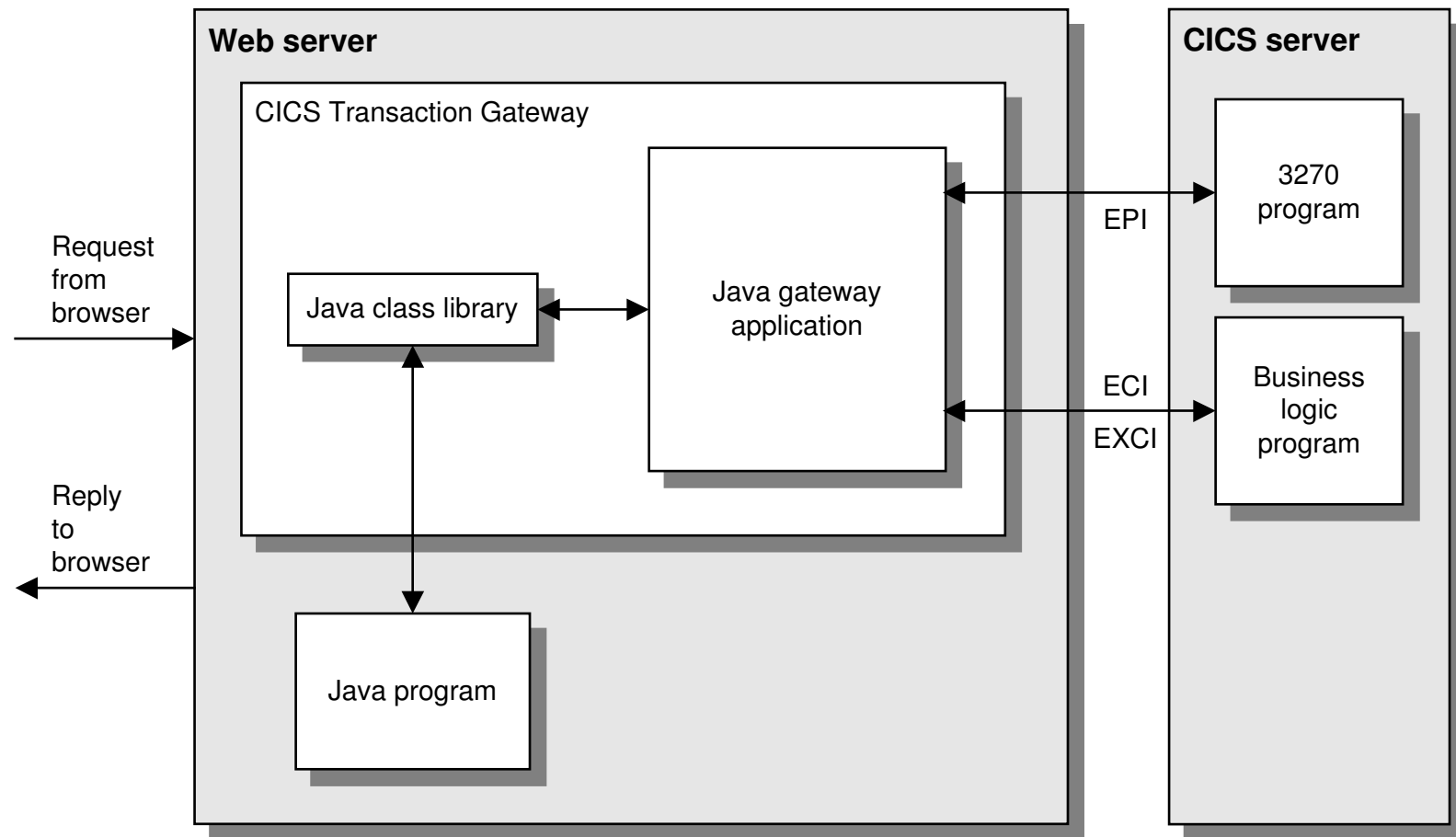
## - Using CICS Web Support

## Techniques for accessing CICS programs from a Web browser

- The services needed to access CICS applications from a web browser can be provided by the *CICS Transaction Gateway* or by *CICS Web Support (CWS)*.
- The CICS Transaction Gateway runs under the control of a web server on either the same OS/390 system as CICS or on a separate server platform.
- CICS Web Support provides the services needed for CICS to function as a web server on its own.



# An overview of the CICS Transaction Gateway



## How the CICS Transaction Gateway works

- The CICS Transaction Gateway includes a *Java gateway application*, which provides services for communicating with CICS programs, and a *Java class library*, which provides the programming interface for using the gateway application services.
- A Java program provides the graphical user interface for a CICS program.
- If a CICS program contains only business logic, the CICS Transaction Gateway can invoke that program using *EXCI* (the *external CICS interface*) when the Gateway is running on the same system as the CICS program or *ECI* (the *External Call Interface*) when it's running on a separate system.
- If the CICS Transaction Gateway is running on a separate system, you can use *EPI* (the *External Presentation Interface*) to communicate with a CICS program that includes 3270 terminal I/O.

## How the 3270 Bridge exit works

- The 3270 Bridge exit lets CICS programs that perform 3270 terminal I/O run on a Web browser with no source code changes.
- The main function of the 3270 Bridge exit is to translate HTML data to 3270 data, and vice versa.
- Before you can use the 3270 Bridge exit with a program, you must modify the BMS mapset for the program with specialized macros. Then, reassembling the mapset creates an HTML version which the 3270 Bridge exit uses to perform the necessary translations.
- The BMS macros for creating HTML documents were introduced with CICS TS 1.3 and provide for some basic formatting of HTML pages.

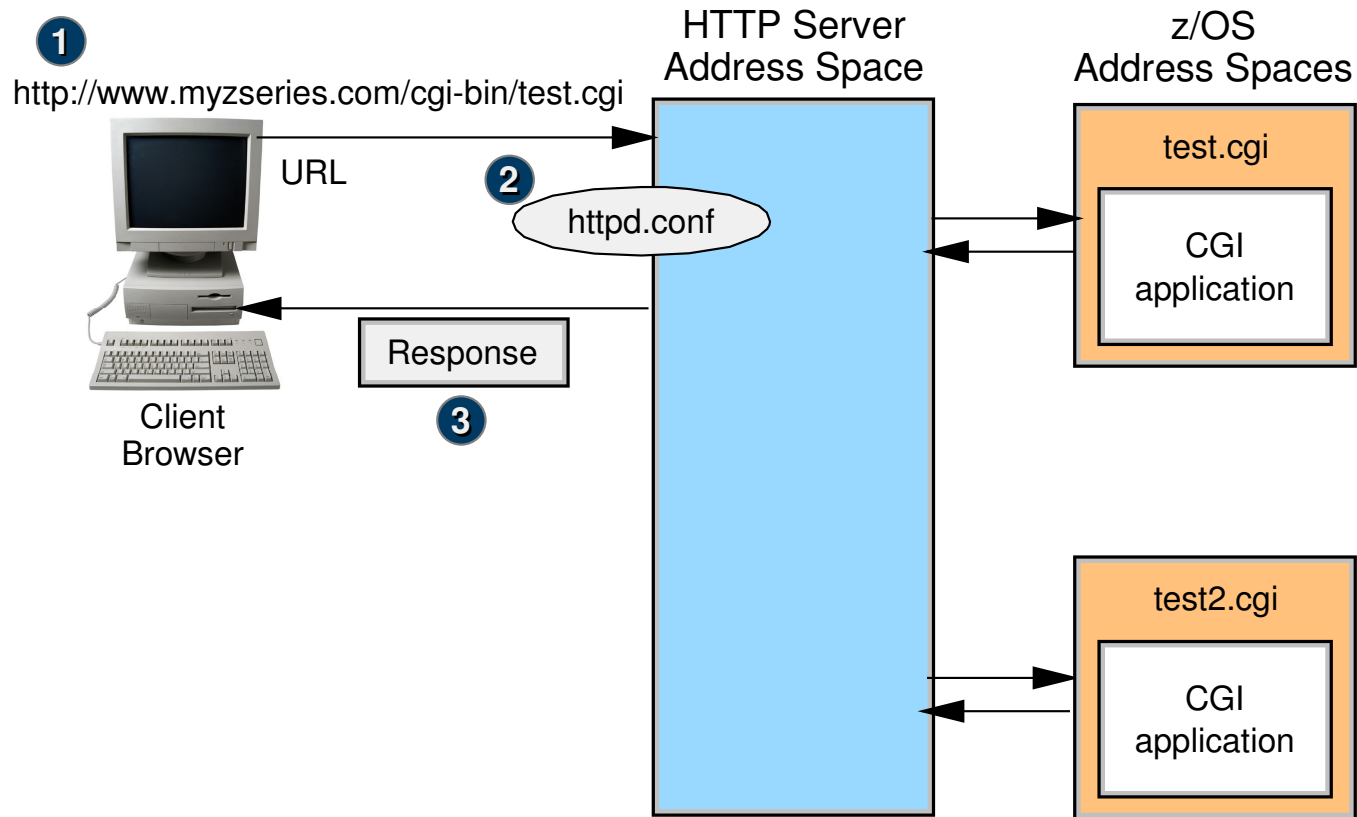
## The customer inquiry screen displayed in a web browser



# Access using Web Services - Using CGI

# Dynamic Web Pages

## Common Gateway Interface (CGI)

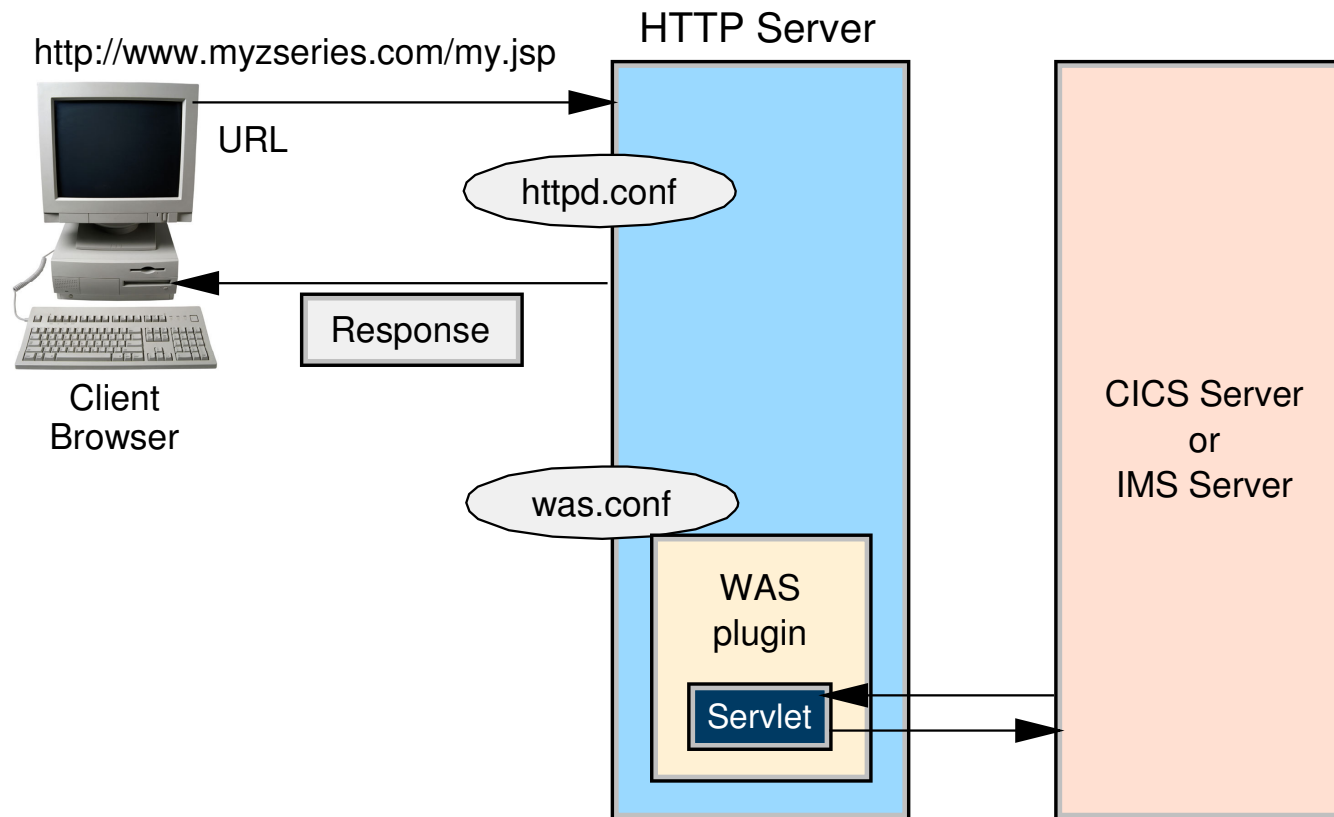


# Access using Web Services

## - Using the WebSphere Application Server / J2EE

# Dynamic Web Pages - Interaction with WebSphere

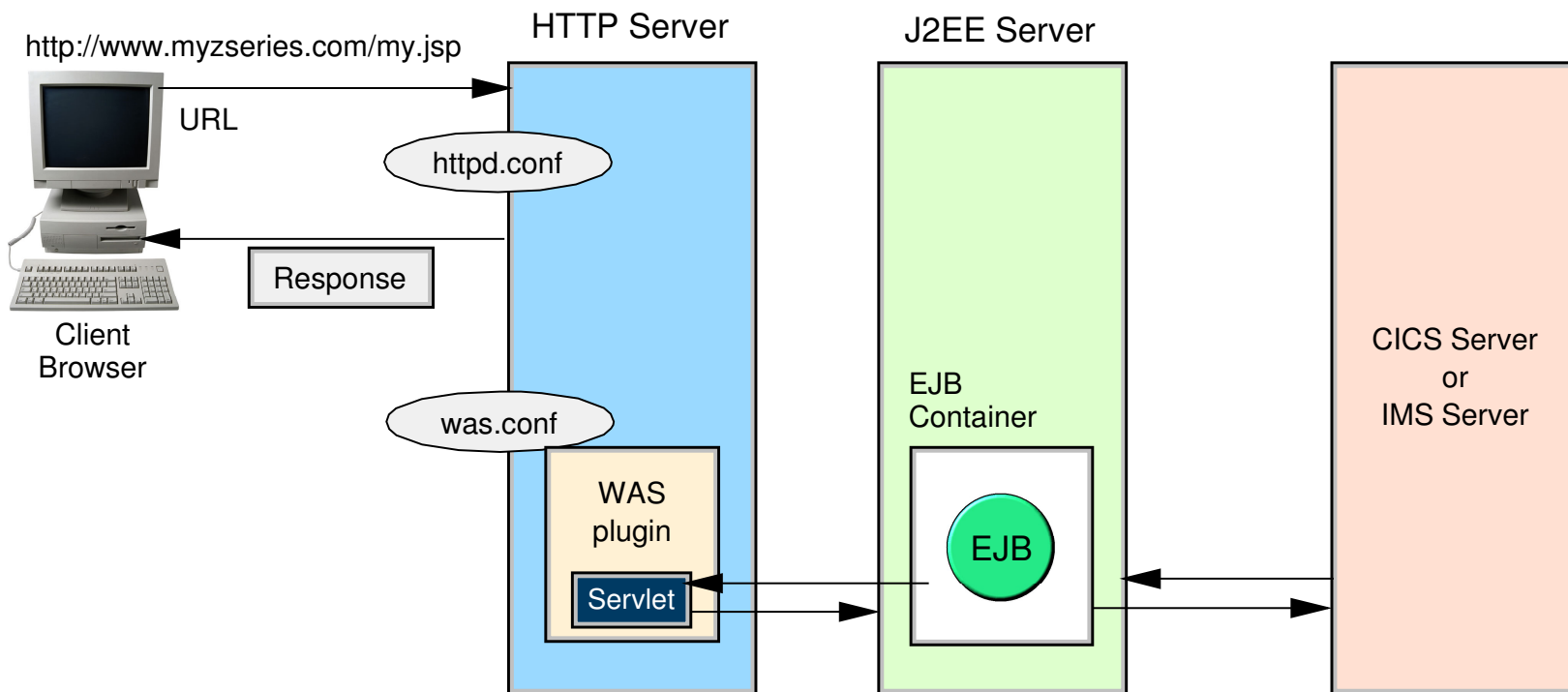
## WebSphere plug-in, same address space





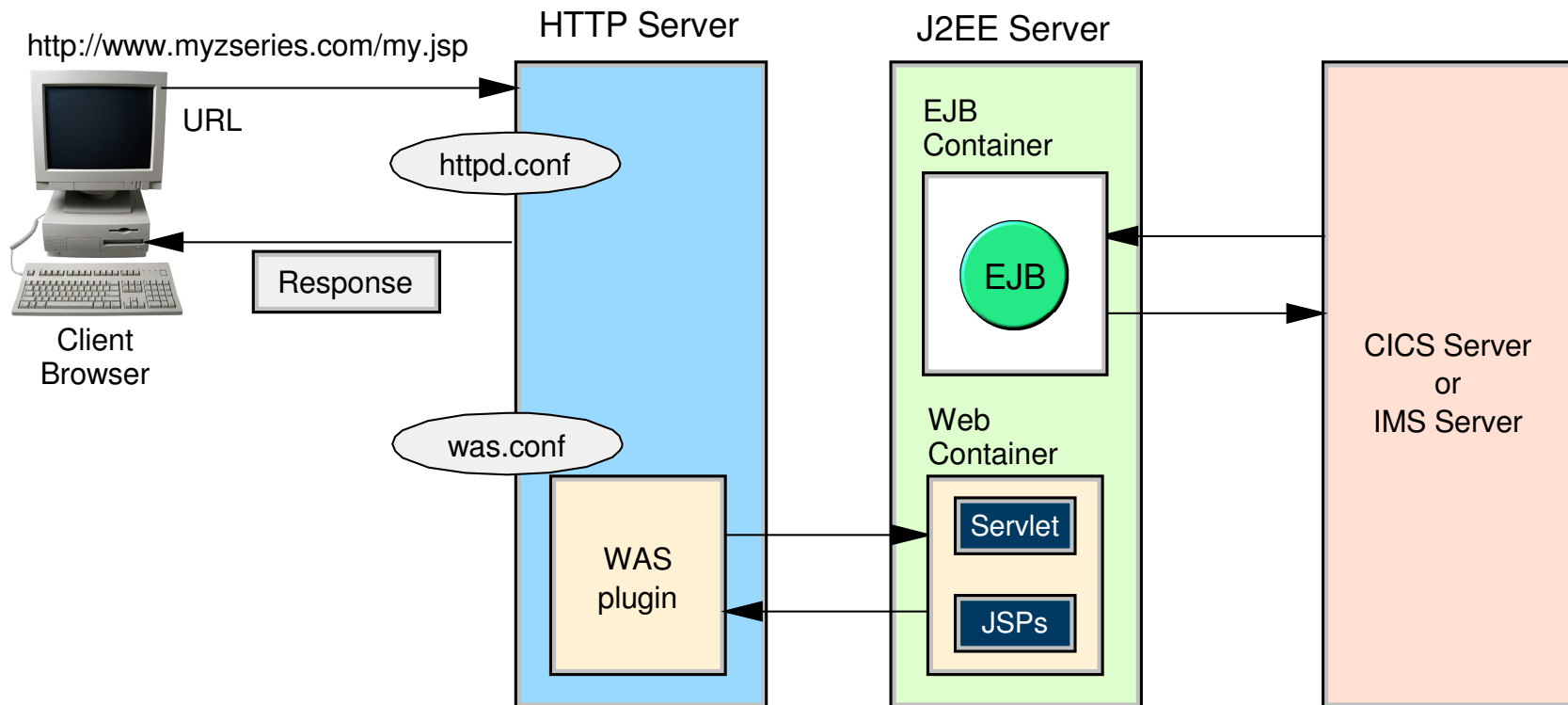
# Dynamic Web Pages - Interaction with WebSphere

**Web container inside HTTP Server, separate EJB container**



# Dynamic Web Pages - Interaction with WebSphere

**Separate J2EE server with both Web container and EJB container**



# J2EE Application Model in z/OS

**Same as on other platforms, following SDK:**

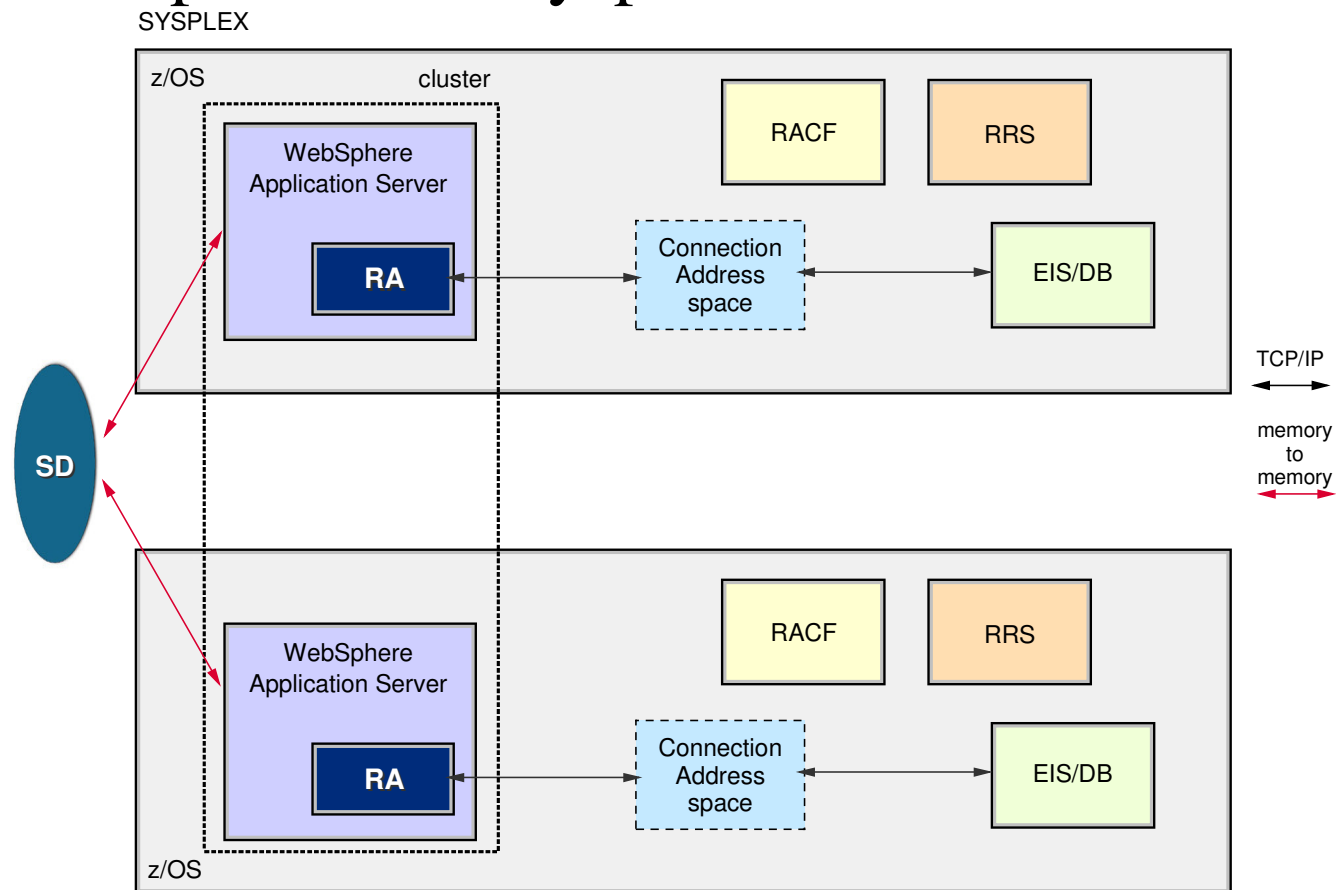
- Functional
- Reliable
- Usable
- Efficient
- Maintainable
- Portable

# Running WebSphere Application Server for z/OS

- Basics of WebSphere on z/OS
- Consolidation of workloads
- WebSphere for z/OS Security
- Continuous availability (see next slide)
- Performance

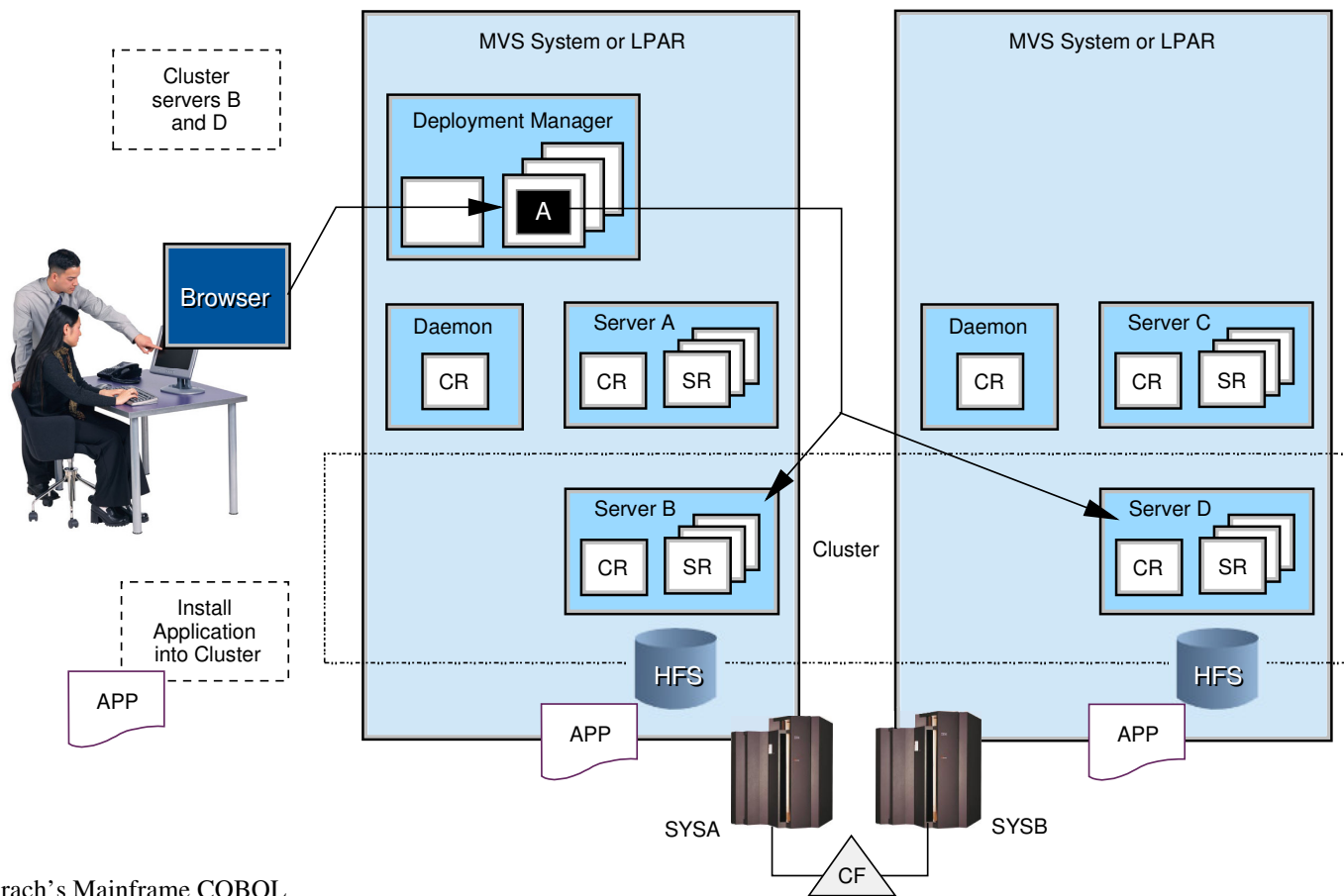
# WebSphere on z/OS - continuous availability

## WebSphere with Sysplex Distributor



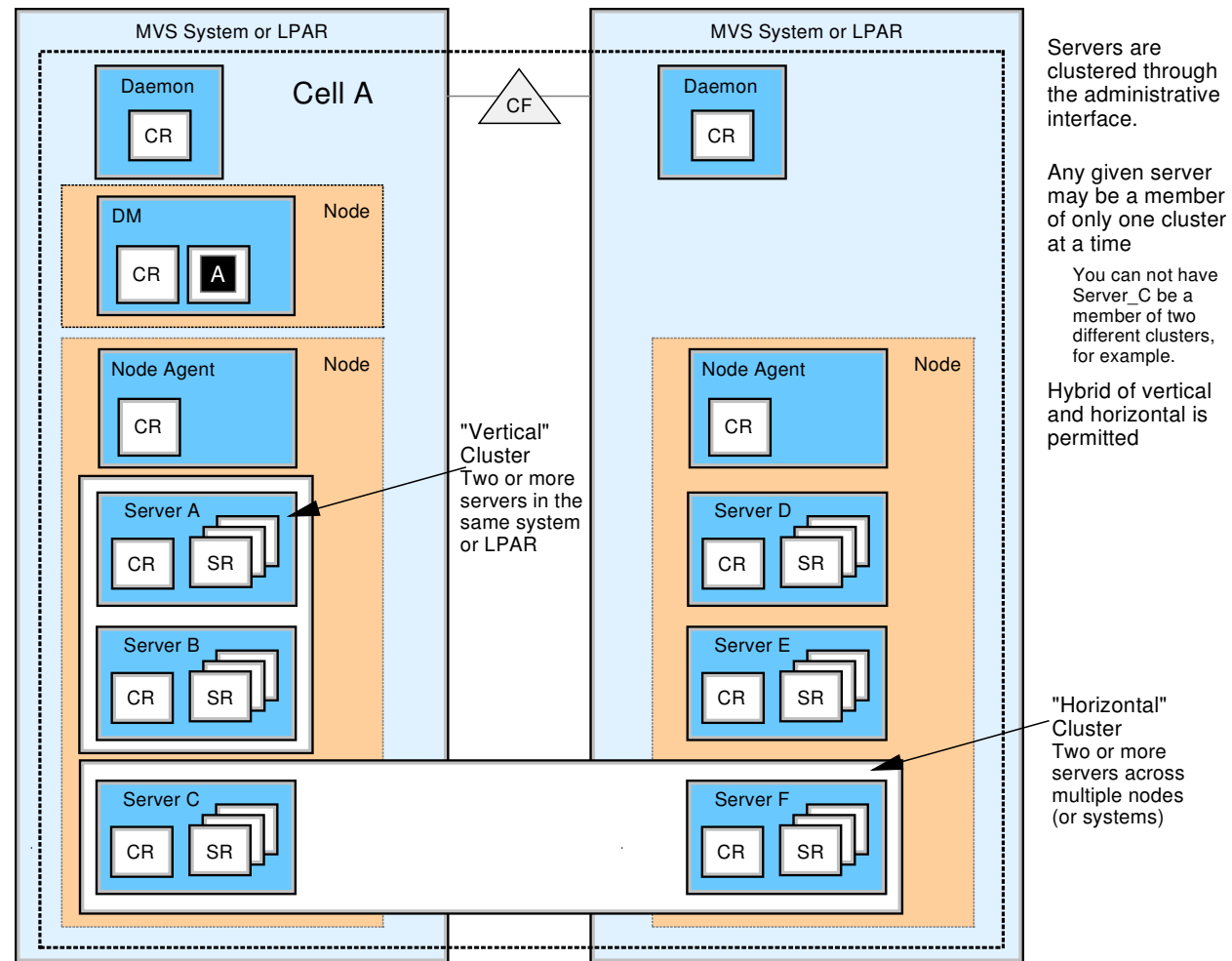
# WebSphere on z/OS - continuous availability

## Clustering of Server in Cell



# WebSphere on z/OS - continuous availability

## Vertical and Horizontal Cluster



# Running WAS on z/OS - Performance

**WebSphere uses three distinct functions of WLM:**

- **Routing**
- **Queuing**
- **Prioritizing**



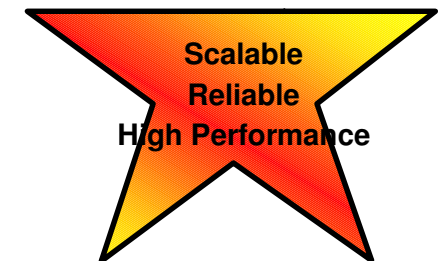
# Accessing IMS Data Web Services & Using SOAP

# IMS Integration Strategy



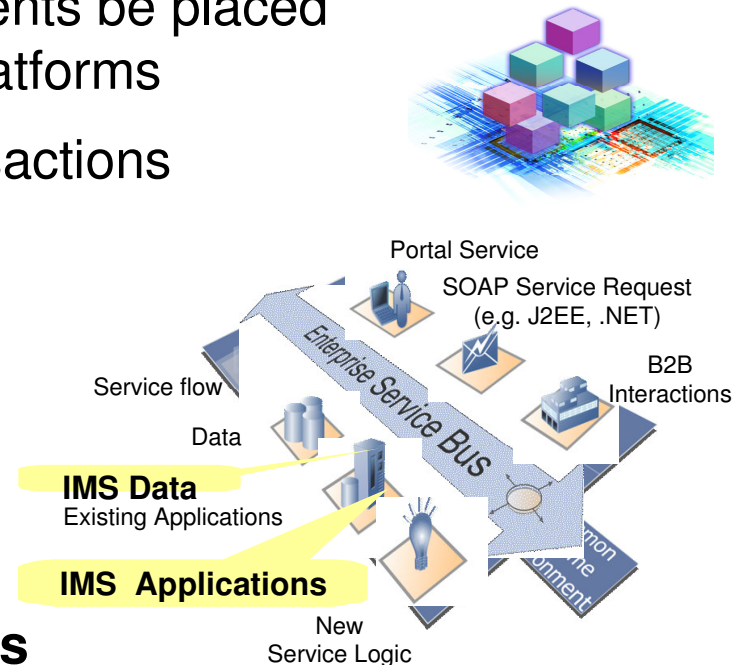
- **Protect customers' investment by enabling integrated access to IMS applications and data**
  - Designed to support open integration technologies
  - Modernize IMS Transactions and Data
- **Enabling tool development/usage to ease application development/enablement**
- **Encourage new application development by supporting standards – XML, SOAP, JAVA**

Murach's Mainframe COBOL



# IMS with Web Services & Service-Oriented Architecture

- **Web services** - allows programmable elements be placed on sites for distributed web access across platforms
- Web services enables unchanged IMS Transactions support a Services Oriented Architecture
  - Leveraging past investments
  - Reducing new programming efforts
  - Aiding business process transformation
- **IMS Transactions enabled as Web services via WebSphere/Rational Servers/Tooling**
- **IMS Transactions enabled as Web services via IMS SOAP Support**



# IMS Application Integration with IMS SOAP support

- **Enables Reuse of IMS Applications as Web Services**
  - Support WebService specifications
  - Leverages open standards and utilizes flexible tooling support
- **Provides IMS Transaction Interoperation with Client Apps**
  - Independent of location, programming language, and platform
  - SOAP Clients can be Microsoft .Net, SAP, Java, etc.

